

# **A hybrid dynamic berth allocation planning problem with fuel costs considerations for container terminal port using Chemical reaction optimization approach**

Arijit De<sup>1</sup>, Saurabh Pratap<sup>2</sup>, Akhilesh Kumar<sup>1\*</sup>, M K Tiwari<sup>1</sup>

<sup>1</sup>Department of Industrial & Systems Engineering, Indian Institute of Technology  
Kharagpur, India

<sup>2</sup>Department of Mechanical Engineering, Indian Institute of Information  
Technology, Design and Manufacturing Jabalpur, India

E-mail id: [arijit.de22@gmail.com](mailto:arijit.de22@gmail.com), [s.pratap@iitkgp@gmail.com](mailto:s.pratap@iitkgp@gmail.com), [akhilesh.niff@iitkgp@gmail.com](mailto:akhilesh.niff@iitkgp@gmail.com),  
[mkt09@hotmail.com](mailto:mkt09@hotmail.com)

## **Abstract**

This paper investigates the dynamic ship berth allocation problem for a container handling port, focusing on vessel waiting time at the anchorage due to the unavailability of the berth and quay cranes. A mixed integer linear programming model (MINLP) considering the fuel cost associated with waiting time and operational time of the docked vessel is developed. The hiring of the quay cranes to load/unload the containers from the ship and arrangement of the vessels in different berths is taken into account. Fuel consumed by the vessels while performing their respective port operations is incorporated in the model for addressing the sustainability aspects in berth allocation problem. A Chemical Reaction Optimization Algorithm (CRO) is proposed to solve the problem in a large-scale realistic environment and compared with the results with Block-Based Genetic Algorithm (BBGA), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The computational experiment illustrates and validates the proposed model on a real case scenario of the port located in India. The case shows that the developed model achieves better utilization of port resources and available berths.

**Keywords:** Container Handling Terminal, Fuel consumption, Meta-heuristic, Berth allocation

## **1. Introduction**

The ports are one of the key components of a Nation's merchandise transport and trading system and since 2001, the total average annual growth of the container terminal port increased by 10%. With 80% of the merchandise trade being carried out by the marine and it reveals that the maritime logistics is the backbone of the world merchandise trade (UNCTAD and WTO, 2015).

In a globalized world, ports form an integral and critical part of the international trade network, as they provide a highly cost efficient movement of cargos across markets. An overwhelming priority of maritime transportation is the movement of containerized cargos from distant suppliers to the consumers. Shipping company uses mega vessels/ships for carrying up to 20,000 twenty-foot equivalent units (TEUs) of containers. These cargo containers (20 foot or 40 foot containers) are loaded onto the ships using port machinery such as quay cranes. The port acts as a buffer between incoming and outgoing vessel traffic as it receives and dispatches some of the containers. Arriving ships and departing rail/road transport facilities serve as physical inputs and outputs for a port. Now, the fierce growth has been observed among the container terminal ports as containerized trade increased by 5.3% in 2014 which is nearly 1.63 billion tons. In order to achieve competitive strategy and customer satisfaction, terminal port must focus on minimizing the turnaround service time of ships by incorporating new operational measure. Moreover, Port operators always aim to optimize the utilization of the port resources for improving the service level and reduce the environmental impact by mitigating the fuel consumption (Hu et al. 2014).

### **1.1. Motivation**

Port operations of a container terminal comprises of seaside operations, land-side operation and yard operations. Seaside operations aim to serve every vessel with a berth for performing its loading/unloading operation at the port. Port operators and ocean carriers share similar objective of reducing the service time of the ships as port operators can provide more berths to other vessels in limited time and ocean carriers can reduce their operating cost which depends on the operating time at the port. For staying competitive, port operators need to develop reliable and accurate berth schedules for avoiding any delay in customer shipment. Berth allocation refers to assigning the berths to certain vessel for loading/unloading processes. In this paper, the berth allocation problem is investigated as berth planning and quay crane assignment are the major part of the port operations to provide smooth ships berthing and departure to container vessel. In reality, dynamic berth allocation is often the bottleneck to enhance the port performance associated with increasing the utilization of port resources such berths and cranes. Thus, the focus of the paper is primarily on complicated port terminal operations which possess significant challenges associated with the delays due to vessel arrival, departure and cargo dispatch through rake/truck. Moreover, many port authorities are concerned about reducing the port-related

pollution and they strive to reduce vessel's fuel consumption in minimizing the environmental pollution. Hence, it is essential to develop some decision making strategies for addressing the port environment protection and vessel fuel consumption.

## **1.2. Contribution**

In this work, a dynamic berth allocation problem is considered aiming to minimize the service time of the berthed vessel at the port. A mathematical formulation is proposed that minimizes the total cost including the fuel cost incurred for the overall service time of all the vessels at the container terminal. The fuel consumption for the vessels performing their port operations is computed for estimating the overall fuel cost incurred. A real case study associated with a container handling port located on the eastern coast of India is presented. As berth allocation problem (BAP) is considered as NP hard (Cordeau et al. 2005, Lu et al. 2012 and Ting et al. 2014), hence exact solution approach cannot resolve large-scale realistic problems associated with BAP. In this paper, chemical reaction optimization (CRO) algorithm is employed for resolving the dynamic berth allocation problem and the results obtained are validated with block-based genetic algorithm (BBGA), genetic algorithm (GA) and particle swarm optimization (PSO) to prove the effectiveness of model. There is no literature available using CRO for talking berth allocation problem, thus it is worthwhile to evaluate CRO for this task. A real case study is presented based on a port located in India to validate the effectiveness of the proposed model in terms of its applicability to a realistic scenario.

In the following section, we summarize the work related to berth allocation. Section 3 describes the problem statement and the mathematical model. In Section 4, we describe the methodology and the approach to solve the problem. The results and discussions are presented in section 5. Finally, we conclude and summarize the work in section 6.

## **2. Literature Review**

Most of the researchers focused on container handling port terminal for the discrete berth and very few scholars studied the problems related to continuous berth container terminal handling port. A brief review of the past literature on the berth allocation problem in the context of container terminal is presented in this section.

## **2.1. Models Depicting Berth Allocation Problem**

Imai et al. (1997) introduced a model for berth allocation and minimizing the ship stay on the port while satisfying the service order of the berth to improve the port utility. The weighted sum method is adopted to formulate the multi-objective model as single objective and determine the solution for both the objectives and satisfying the tradeoff. Later, Imai et al. (2001) developed a heuristic to solve the dynamic discrete berth allocation problem for the public port. The model is formulated to handle (1) the static berth allocation and then extended to (2) dynamic discrete berth allocation considering minimization of the ship waiting and operational time. Other authors also presented different extensions of berth allocation problem such as Kim et al. (2003) dealt with berth scheduling problem and allocation of containers at the wharf. The mixed integer programming is proposed to capture the berthing operations and determine the loading/unloading time of containers. A variant of berth allocation problem addressed by Guan and Cheung (2004) aiming to minimize the service time of berthed vessel. The heuristic approach in their work is based on random search and tree search used for determining the optimal ships berthing position at the port. The proposed approach provides the optimal berthing space for berthing at the terminal port and minimizes the operational cost/dispatch cost. Several researchers employed different meta-heuristic algorithms like variable neighborhood search algorithm for resolving a berth allocation problem and determining the cost for ship waiting and handling time (Hansen et al. (2008)).

Problem variation can be found with respect to integrating berth allocation with quay crane assignment for a container terminal handling port as presented by Iris et al. (2015) and set partitioning method is employed to minimize the costs incurred from the vessel handling and the quay crane allocation at the port. In terms of joint planning problem, Tao and Lee (2015) focused on a berth and yard allocation problem for transshipment hubs and aiming to minimize the total distance of exchanging containers between mother vessels and feeders. Few authors such as Hendriks et al. (2013) and Zhen (2015) dealt with a tactical berth allocation problem for explicitly addressing the uncertainty in berthing activities and reducing the operational dwell time. Several other researchers integrated quay crane allocation problem with discrete berth allocation for minimizing the ship operational time (Pratap et al (2015a), and Wu and Ma (2017)).

Although it has been observed that discrete berth allocation problem attracted a large attention in academic researches. Some of the research papers considered the discrete berth allocation for port terminal such as Pratap et al. (2015b) proposed a discrete berth allocation problem and reduced the ship operational time and ship berthing order, while Oliveira et al. (2014) dealt with a discrete berth allocation problem for minimizing the service time of the berthed ship, whereas Ribeiro et al. (2016) dealt with a discrete berth allocation problem for mitigating the demurrage cost and resolving the complexities of vessel selection at the anchorage. For the discrete berth allocation problems, the quay is partitioned into a number of berth segments where only one vessel could be handled at a time. For continuous berth allocation problem, vessel could be berthed wherever any empty spaces are available. Babu et al. (2015) proposed rule based heuristic approach to solve the discrete berth allocation problem for the cargo handling port.

## **2.2. Solution Approaches**

The berth allocation problem (BAP) is a NP-hard problem as stated by several researchers such as Lim et al. (1998), Cordeau et al. (2005), Lu et al. (2012), Ting et al. (2014) and Pratap et al. (2015a). Due to the computational complexity associated with resolving the berth allocation problem, several researchers have proposed intelligent search methods and random search techniques. These heuristic methods include simulated annealing employed by Kim et al. (2003) and Moorthy and Teo (2006) for solving berth scheduling problem. Several researchers such as Nishimura et al. (2001), Imai et al. (2007), Lalla et al. (2014) and Lalla et al. (2016) used genetic algorithm in the context of berth allocation problem and obtained considerably good solution. Cordeau et al. (2005) stated berth allocation problem as a NP-Hard problem and developed a tabu search algorithm for dealing with a discrete and continuous case for BAP. Algorithms such as ant colony optimization (Tong et al. (1999) and Cheong et al. (2008)) and particle swarm optimization (Ting et al. 2014) are employed to resolve berth allocation problem. Other researchers employed with variable neighborhood search (Hansen et al. (2008)), adaptive large neighborhood search (Mauri et al. (2016)) and greedy randomized adaptive search heuristic (Lee et al. 2010) to obtain near-optimal solution while dealing with a berth allocation problem. In this paper, a chemical reaction optimization (CRO) algorithm is presented to solve dynamic berth allocation problem for a container terminal. The reasons for using CRO in order to resolve the problem are as follows. CRO is employed for solving well-known NP-hard problems including

Quadratic Assignment Problem (QAP), and Channel Assignment Problem (CAP) and Resource-Constrained Project Scheduling Problem (RCPSP). Furthermore, the effectiveness of CRO algorithm over other metaheuristics is showed by solving different optimization problems. CRO algorithm has been widely employed for different varieties of problems such as job-shop scheduling justifying its ability to deal with complex combinatorial problems in different domains (Li and Pan (2012)). Other NP-hard problem such as 0-1 Knapsack problem (KP01) which is a well-known combinatorial optimization problem has been resolved using a new chemical reaction optimization with greedy search algorithm (Truong et al. 2013). Several researchers such as Roy et al. (2014) and Li et al. (2014) proposed the hybridization of chemical reaction optimization algorithm in their paper with differential evolution and variable neighborhood search respectively. CRO has been widely used to resolve multi-objective problems and Li et al. (2015) presented a hybrid algorithm combining particle swarm optimization with CRO for multi-objective optimization. Bhattacharjee et al. (2014) provided a real coded version of chemical reaction optimization algorithm and validated with several existing optimization techniques to justify the superiority of the algorithm in terms of solution quality and computational efficiency. As several researchers have employed CRO algorithm and showcased its superiority in terms of solution quality and computational efficiency over other contemporary algorithms, hence CRO is used in this paper to resolve dynamic berth allocation problem.

### **2.3. Research Gaps**

Even though CRO algorithm has been used to deal with different combinatorial problems, but there is no research of CRO algorithm employed to solve BAP. Hence, the application of CRO in resolving a dynamic ship berth allocation problem for a terminal port is one of the contributions of the paper. Contemporary algorithms such as genetic algorithm, particle swarm optimization and ant colony optimization are applied on the problem studied in this paper to validate the result obtained from chemical reaction optimization algorithm.

Majority of the berth allocation problem studied earlier in the literature are either discrete or continuous cases of BAP. Very few researchers such as Nishimura et al. (2001), Cordeau et al. (2005), Moorthy and Teo (2006), Imai et al. (2007), Cheong and Tan (2010) and Umang et al.

(2013) studied dynamic hybrid berth allocation problem yet overlooking the aspects of quay crane allocation to each berthed vessel while dealing with a BAP. To bridge this gap in the literature, the problem of dynamic hybrid berth allocation problem is presented and merged with quay crane allocation. The problem is solved for a large scale realistic environment based on a real data of a port located in India. Moreover, only a few studies considered fuel consumption by vessels while addressing a berth allocation problem (Golias et al. (2009), Lang et al. (2010) and Hu et al. (2014)). Hence, it is essential to bridge this gap and the current study incorporates the sustainability aspects within dynamic hybrid berth allocation problem.

### **3. Problem Description**

In dynamic hybrid berth allocation, the quay is partitioned into sets of segments and a vessel can occupy more than one segment at a time or multiple vessels can be allowed to share the same quay at a time. Whereas for discrete berth allocation, the quay is divided into berths segments and one vessel can occupy a single berth segment. The graphical representation of the hybrid berth allocation is shown in figure 1. A set of vessels needed to be berthed on a discrete quay of a specific length for a specific time horizon. The quay boundary is discretized into set of segments and vessels occupy more than one segments at a given time. As in a container terminal the cargo is packed into containers, hence only quay crane is used to handle the containers for loading and unloading purpose. The clearance distances between adjacent vessels are considered to avoid any overlapping in space.

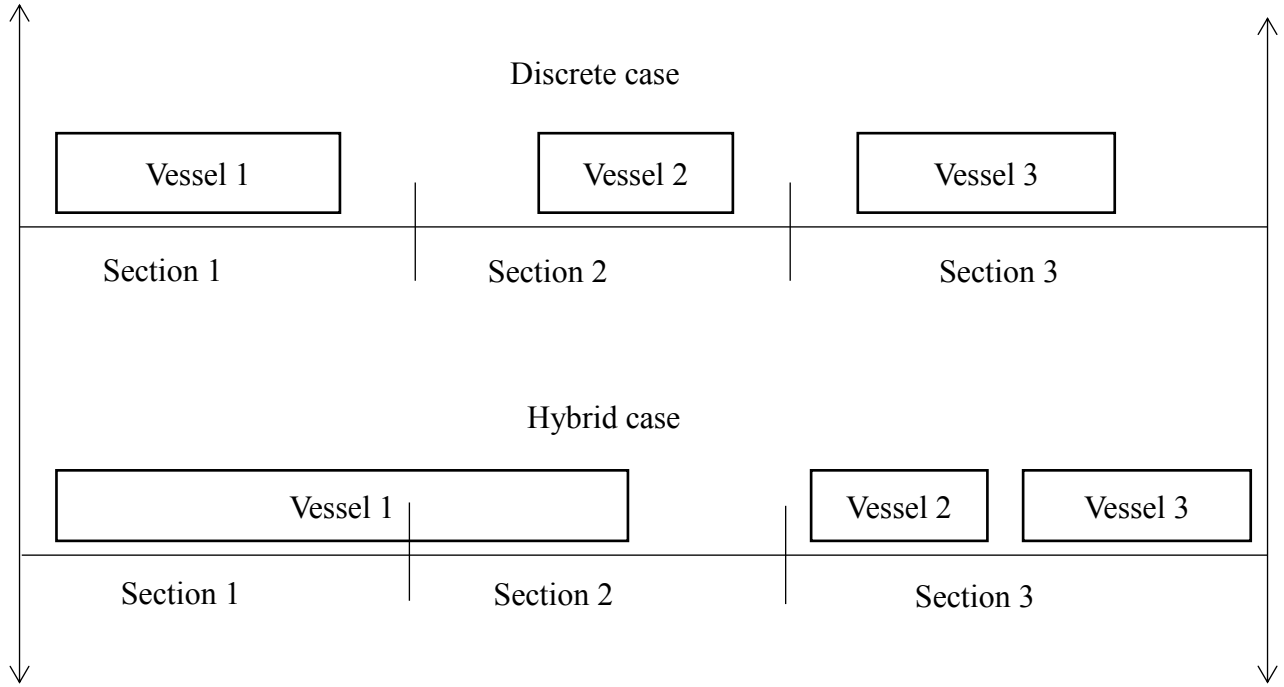


Figure 1: Discrete and hybrid berth allocation

Figure 2a shows the two-dimensional (2-D) view of a port terminal considering berth segments. The port consists of a berth of  $m$  segments and stockyards to load/unload the containers to the docked vessel. The port operator assigned the tug boats to guide the ship from anchorage to the available berth segments  $m$ . The available total  $m$  segment on the berth should not be less than the length of the berthed ship and the quay cranes need to be assigned for carrying out the loading/unloading of the cargo on the ship from storage area at the port terminal. A mathematical model pertaining to dynamic hybrid berth allocation problem is proposed for minimizing the operational cost and fuel cost of the ship related to service time and waiting time. Total fuel consumed by vessels is determined for estimating the overall fuel cost incurred. The clearance distance concerning two adjacent vessels needs to be implicitly taken into account along with end-clearances in vessel lengths for avoiding the any risk between two successive vessels. There need to be appropriate clearance times between the handling times of two successive vessels for avoiding any overlapping. Figure 2b shows a feasible berth allocation solution without any overlapping of the successive vessel.



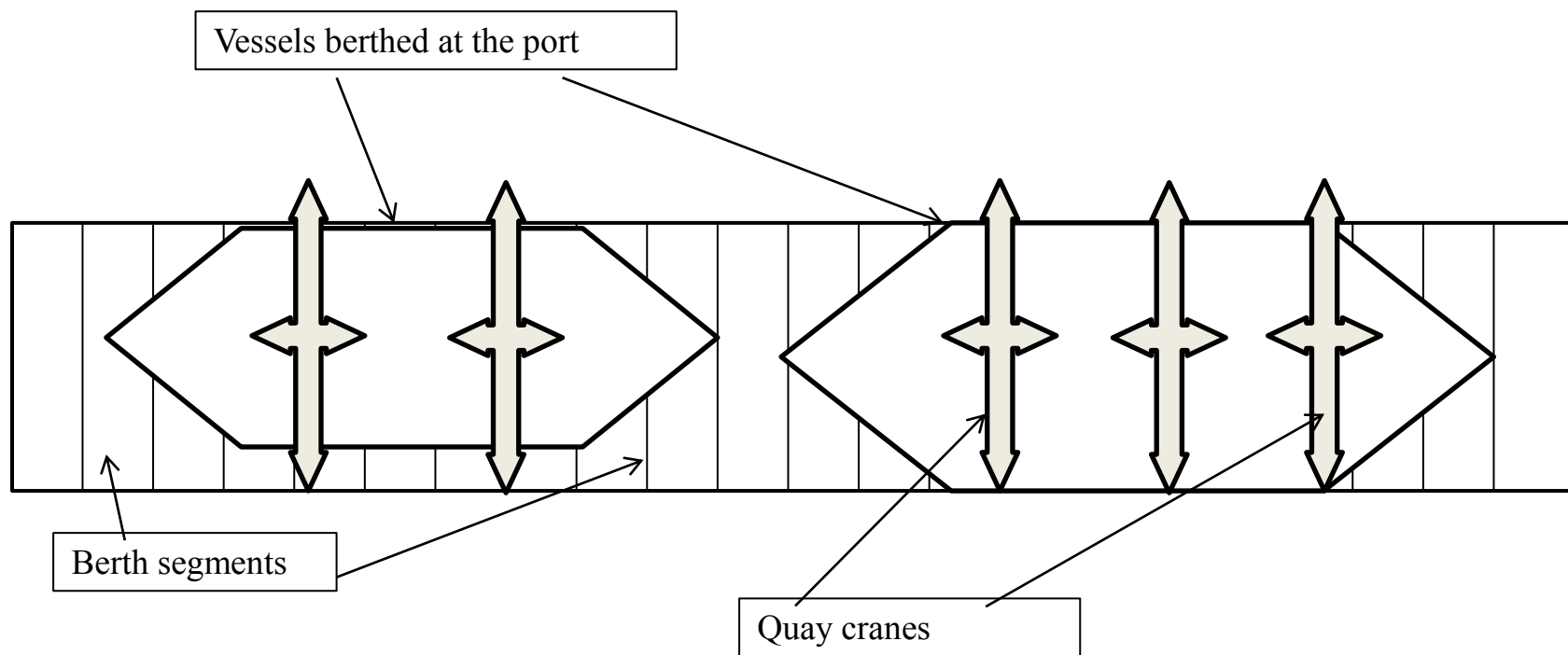


Figure 2a: Non-overlapping of two successive vessels on berth segments

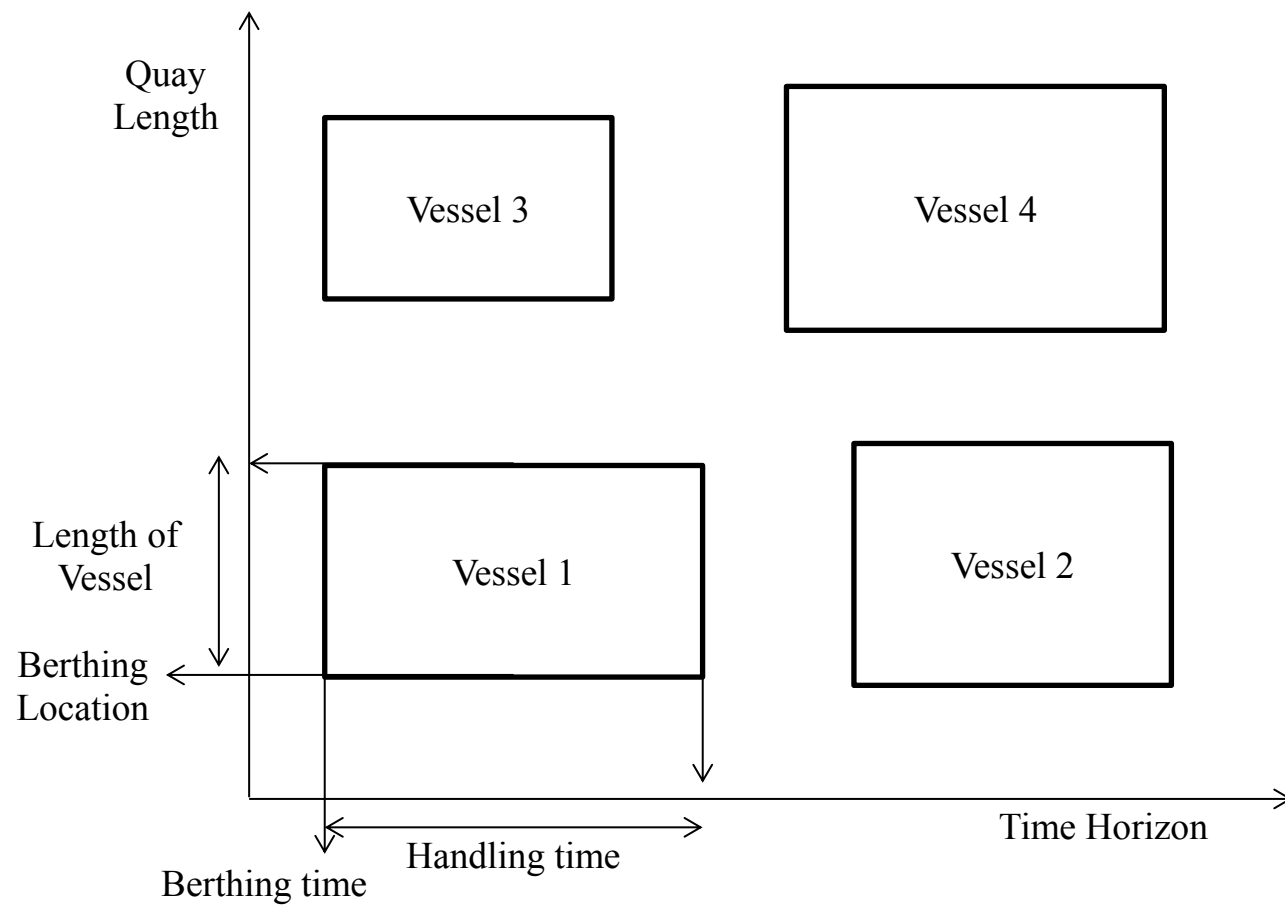


Figure 2b: Feasible berth allocation solution

### 3.1. Mathematical Model

#### Indices

$v, s$	Vessel
$m, n$	Segment
$c$	Container group
$t$	Time period

#### Sets

$V$	Sets of vessels
$M$	Sets of segments
$C$	Sets of container group
$T$	Sets of time period

#### Parameters

$\alpha_c^t$	Time required in handling a container of block $c$ in time period $t$
$\beta_v$	Maximum service time provide to vessel $v$
$L_v$	Length of vessel $v$ in terms of number of segments it can occupy
$W_v$	Draft of the vessel $v$
$\rho_m$	Draft of segment $m$
$E_{vt}$	Expected arrival time of vessel $v$ in time period $t$
$A_{vt}$	Expected departure time of vessel $v$ in time period $t$
$b_m$	Starting coordinate of segment $m$
$\underline{a_{vt}}, \overline{a_{vt}}$	Earliest and latest arrival time of vessel $v$ in time period $t$
$L$	Total number of segments of the quay
$P_{vc}^t$	Container group $c$ to be loaded or unloaded on vessel $v$ in time period $t$
$C_{vt}^1$	Penalty cost per hour for delaying the departure of vessel $v$ in time period $t$

$C_{vt}^2$	Penalty cost per hour of vessel $v$ in time period $t$ for waiting before starting the operation
$F_v^{port}$	Fuel consumed (tons per hour) at the port by vessel $v$
$F^{max}$	Maximum fuel consumed while operating after its expected departure time
$C^{fuel}$	Price of fuel (Rs./ton)
$B$	Large positive constant
$H^t$	Fixed cost for hiring a quay crane in time period $t$
$Q_v$	Maximum number of quay crane that can be assigned to vessel $v$
$R_c$	Upper bound on the number of container group $c$ handled by a quay crane

### ***Decision Variables***

$f_v^t$	Starting time of handling vessel $v$ in time period $t$
$h_v^t$	Total handling time of vessel $v$ in time period $t$
$a_v^t$	Arrival time of vessel $v$ in time period $t$
$d_v^t$	Departure time of vessel $v$ in time period $t$
$q_v^t$	Number of quay crane assigned to vessel $v$ in time period $t$
$\lambda_c^t$	Number of container group $c$ handled by a quay crane in time period $t$
$x_{vmt}$	$\begin{cases} 1 & \text{if vessel } v \text{ occupies berth segment } m \text{ in time period } t \\ 0 & \text{otherwise} \end{cases}$
$\theta_{vm}^t$	$\begin{cases} 1 & \text{if segment } m \text{ is the starting segment of vessel } v \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$
$\delta_{sv}^t$	$\begin{cases} 1 & \text{if vessel } v \text{ is berthed to the left of vessel } s \text{ without any overlapping in period } t \\ 0 & \text{otherwise} \end{cases}$
$\omega_{sv}^t$	$\begin{cases} 1 & \text{if vessel } v \text{ is positioned earlier than vessel } s \text{ in time period } t \\ 0 & \text{otherwise} \end{cases}$
$z_{sv}^t$	$\begin{cases} 1 & \text{if handling of vessel } v \text{ finishes before handling of vessel } s \text{ in time period } t \\ 0 & \text{otherwise} \end{cases}$

$$y_{vmn}^t = \begin{cases} 1 & \text{if vessel } v \text{ starting at segment } m \text{ touches segment } n \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

### Objective Function

$$\text{Minimize } \sum_{v \in V} \sum_{t \in T} \left[ \left( C_{vt}^1 + F_v^{port} C^{fuel} \right) \left\{ \text{Max} \left( E_{vt} - a_v^t, 0 \right) \right\} + \left( C_{vt}^2 + F_v^{port} C^{fuel} \right) \left( d_v^t - f_v^t \right) + H^t q_v^t \right] \quad (1)$$

Equation (1) presents the objective function of the model consisting of four terms. The first term is related to the cost associated with the waiting before starting port operation and fuel cost incurred while waiting. The second term depicts the cost related to the total service time and the total fuel cost incurred while operating at the port. The third term presents the fixed cost for hiring quay cranes.

### Constraints

$$f_v^t - E_{vt} \geq 0, \quad \forall v \in V, \forall t \in T \quad (2)$$

$$f_v^t \geq a_v^t, \quad \forall v \in V, \forall t \in T \quad (3)$$

$$\underline{a}_{vt} \leq a_v^t \leq \overline{a}_{vt}, \quad \forall v \in V, \forall t \in T \quad (4)$$

$$h_v^t = \frac{\sum_{c \in C} P_{vc}^t \alpha_c^t}{q_v^t} \quad \forall v \in V, \forall t \in T \quad (5)$$

$$\sum_{t \in T} h_v^t \leq \beta_v \quad \forall v \in V \quad (6)$$

$$q_v^t = \sum_{c \in C} \frac{P_{vc}^t}{\lambda_c^t} \quad \forall v \in V, \forall t \in T \quad (7)$$

$$q_v^t \leq Q_v, \quad \forall v \in V, \forall t \in T \quad (8)$$

$$\lambda_c^t \leq R_c, \quad \forall c \in C, \forall t \in T \quad (9)$$

$$f_v^t + h_v^t \leq d_v^t, \quad \forall v \in V, \forall t \in T \quad (10)$$

Constraint (2) ensures the servicing of the vessel only after its expected arrival at the port. Constraint (3) states that the vessel can start its port operation only after its arrival. Constraint (4) provides the arrival time range for the vessel at a port. Constraint (5) depicts the handling time of

the vessel considering the number containers to be loaded or unloaded and number of quay cranes working on the vessel simultaneously. Constraint (6) makes sure that the total handling time of the vessel should be less than or equal to its maximum service time. Constraint (7) depicts the number of quay cranes required by a vessel to perform its port operation and constraint (8) keeps an upper bound on the number of quay cranes that can be assigned to a vessel. Constraint (9) provides an upper bound on the number of container of different groups that can be handled by a quay crane. Constraint (10) presents the relationship between the departure time of vessel with that of the handling time and arrival time of the vessel.

$$d_v^t \leq f_s^t + B(1 - \omega_{sv}^t), \quad \forall v, s \in V, \forall t \in T \quad (11)$$

$$f_s^t + B(1 - z_{sv}^t) \geq f_v^t + h_v^t, \quad \forall v, s \in V, \forall t \in T \quad (12)$$

$$\sum_{m \in M} \theta_{sm}^t b_m + B(1 - \delta_{sv}^t) \geq \sum_{m \in M} \theta_{vm}^t b_n + L_v, \quad \forall v, s \in V, \forall t \in T \quad (13)$$

$$\delta_{sv}^t + \delta_{vs}^t + z_{sv}^t + z_{vs}^t \geq 1, \quad \forall v, s \in V, \forall t \in T \quad (14)$$

Constraint (11) – (14) presents the non-overlapping restrictions for any two vessels berthing at the port. Constraints (11), (12) and (13) are linearized by using a large positive constant value  $B$ .

$$\sum_{m \in M} \theta_{vm}^t = 1, \quad \forall v \in V, \forall t \in T \quad (15)$$

$$\sum_{m \in M} \theta_{vm}^t b_m + L_v \leq L, \quad \forall v \in V, \forall t \in T \quad (16)$$

$$\sum_{m \in M} (y_{vmn}^t \theta_{vm}^t) = x_{vnt} \quad \forall v \in V, \forall t \in T, \forall n \in M \quad (17)$$

$$\text{If } \theta_{vm}^t = 1, \quad \forall v \in V, \forall t \in T, \forall m \in M \quad (18)$$

$$\text{then } y_{vmn}^t = 1, \text{ for } n = m + 1 \dots m + L_v$$

$$(\rho_m - W_v) x_{vnt} \geq 0, \quad \forall v \in V, \forall t \in T, \forall n \in M \quad (19)$$

$$F_v^{port} \left[ \text{Max}(d_v^t - A_v, 0) \right] \leq F^{max} \quad \forall v \in V, \forall t \in T \quad (20)$$

Constraint (15) provides the information about the starting segment of the berth occupied by the vessel. Constraint (16) ensures that the length of the vessel should be less than the length of the quay. Constraint (17) provides the information about the segments occupied by the vessel and

constraint (18) ensures that the number of segments occupied by a vessel should be equivalent to the length of the vessel. Constraint (19) states that the draft of the vessel should be less than the draft of the segment occupied by the vessel. Constraint (20) ensures that the fuel consumed by the vessel while operating after its expected departure from the port should always be less than the maximum amount of the fuel that can be consumed.

$$f_v^t, h_v^t, a_v^t, d_v^t, q_v^t \geq 0 \quad \forall v \in V, \forall t \in T \quad (21)$$

$$\lambda_c^t \geq 0 \quad \forall t \in T, \forall c \in C \quad (22)$$

$$x_{vmt}, \theta_{vm}^t \in \{0,1\}, \quad \forall v \in V, \forall t \in T, \forall m \in M \quad (23)$$

$$\delta_{sv}^t, \omega_{sv}^t, z_{sv}^t \in \{0,1\}, \quad \forall v, s \in V, \forall t \in T \quad (24)$$

$$y_{vmn}^t \in \{0,1\}, \quad \forall v \in V, \forall t \in T, \forall m, n \in M \quad (25)$$

Equations (21) and (22) represent the non-negativity constraints. Equations (23), (24) and (25) depict the binary variables. Berth allocation problem is more difficult to solve due to its NP-hard nature as mentioned by Lim et al. (1998), Cordeau et al. (2005), Lu et al. (2012), Ting et al. (2014) and Pratap et al. (2015a). Hybrid discrete berth allocation problem is more complicated version of the BAP and solving such a problem using exact solution algorithm needs excessive memory requirements and huge computational time. Although, the mathematical model aims to solve a large-size problem instance depicting a real-life case study of a port located in India. Several researchers have developed various random search optimization techniques like genetic algorithm, simulated annealing, particle swarm optimization for resolving NP-hard problems. In the same area, chemical reaction optimization algorithm inspired from the chemical reaction of molecules is becoming popular in dealing with real-life NP-hard problems (Lam et al. (2012) and Troung et al. (2013)). As chemical reaction optimization (CRO) is a powerful algorithm in resolving combinatorial optimization problems, hence CRO is employed to solve the mathematical formulation proposed earlier.

#### 4. Solution Methodology

The computation of the proposed mathematical model is challenging as the problem sizes increases exponentially with the increase in the number of vessels, berth segments, time period and container groups. Exact optimization algorithms such as branch and bound algorithm, branch

and price algorithm etc. can be used to solve linear as well as non-linear problems. Although, such algorithms require huge memory space and significant computational time for solving even a medium size problem instances (Guan and Lin (2016), De et al. (2016)). Hence, for dealing with large-size problem instances, meta-heuristic techniques have been chosen for obtaining near-optimal solution within less computational time (De et al. (2015) and De et al. (2017b)). Therefore, for resolve the hybrid dynamic berth allocation problem a chemical reaction optimization algorithm is presented.

Chemical reaction optimization (CRO) algorithm is inspired from the two laws of thermodynamics - conservation of energy and entropy of the system always increases. The structural changes of the molecules in chemical reaction are activated by the collision occurring between multiple molecules or between molecules and the wall of the container. CRO is a recently transformed nature inspired meta-heuristic algorithm depicting the interaction of molecules in the form of a chemical reaction to obtain the global optimal solution for discrete optimization. CRO is governed by four elementary reactions – two unimolecular reactions and two inter-molecular reactions. On-wall ineffective collision and decomposition belongs to the category of on-wall ineffective collision. Inter-molecular ineffective collision and synthesis are associated with inter-molecular category. In unimolecular reactions, a single molecule may collide with the wall of the container and might decompose into two molecules or remain as single molecule. For inter-molecular case, two molecules collide with each other to form a single molecule or may remain as two molecules. On-wall ineffective collision and inter-molecular ineffective collision perform the exploitation (local search) of the algorithm and the exploration is carried out by the decomposition and the synthesis reactions. The exploitation and exploration strategies are used simultaneously to find the global optima within the solution space.

Chemical reaction optimization algorithm is a variable population based evolutionary algorithm where molecule acts as a manipulating agents. The molecule structure ( $\omega$ ) depicts the variable arrangement of the mathematical formulation, whereas potential energy ( $PE$ ) provides the objective function value corresponding the molecule structure and kinetic energy ( $KE$ ) is the measure of tolerance for accepting the worst solutions. Other features of a CRO algorithm are number of hits ( $NumHit$ ) and minimum structure ( $MinStruct$ ) correspond to the total number of collisions experienced by the molecule and molecule structure experiencing minimum



potential energy respectively. The potential energy of the molecule attaining *MinStruct* is referred to as minimum potential energy (*MinPE*) and the number of moves taken by a molecule in realizing the minimum structure (*MinStruct*) is represented as minimum hit number (*MinHit*).

#### 4.1. Initialization Phase

Chemical reaction optimization algorithm requires an initial solution comprising of the values of the decision variables of the mathematical model and this initial solution is fed into the molecules of the CRO algorithm. The initial solution is obtained by generating the values of the decision variables using some of the equations and satisfying majority of the constraints for retaining the feasibility of the solution. The variable  $q_v^t$ , associated with the number of quay crane assigned to a vessel is generated by considering the upper bound on the number of quay cranes for each vessel given by equation (8). The integer variable  $\lambda_c^t$ , related to the number of container groups handled by a quay crane is generated using the value of variable  $q_v^t$  and equation (7) and the obtained value of the variable  $\lambda_c^t$  is used to satisfy the equation (9). Using the value of the variable  $q_v^t$  and equation (5), the variable  $h_v^t$  associated with the handling time of a specific vessel can be determined. The value obtained for the variable  $h_v^t$  is used to satisfy constraint (6). Arrival time of a vessel  $a_v^t$  is generated using the earliest and latest arrival time of a vessel given in equation (4). Using the value of the variable  $a_v^t$  and equations (2) and (3), the starting time of handling vessel,  $f_v^t$  can be obtained. The departure time of the vessel  $d_v^t$  can be generated by employing equation (10) and using the starting time of handling for a vessel and total handling time of the vessel. The value of the binary variable  $\omega_{sv}^t$  related to the positioning of two vessels is obtained using equation (11) and values of the variables associated with the departure time of one vessel  $d_v^t$  and starting time of port operation for another vessel  $f_s^t$ . The value of the binary variable,  $z_{sv}^t$  depicting the relationship between the handling of one vessel with respect to handling of another vessel is obtained using equation (12) and values of the variables related to the start time of port operation for one vessel and start time and total handling time of another vessel. Initially, the value of the binary variable  $\theta_{vm}^t$ , representing the

starting segment of a vessel is obtained using equation (15) and (16). The value obtained for the binary variable  $\theta_{vm}^t$  is fed into equation (13) to obtain the value of the binary variable  $\delta_{sv}^t$  while satisfying the non-overlapping restrictions between two vessels. Values obtained for binary variables  $\delta_{sv}^t$  and  $z_{sv}^t$  are employed to satisfy the constraint (14). The specific sections occupied by a vessel can be depicted by determining the value of the binary variable  $y_{vmn}^t$  from the equation (18) using the value associated with variable  $\theta_{vm}^t$  representing vessel's starting segment. The value of the binary variables  $y_{vmn}^t$  and  $\theta_{vm}^t$  can be fed in equation (17) to compute the segments occupied by the vessel or the binary variable  $x_{vmt}$ . The value obtained for the binary variable  $x_{vmt}$  is used to satisfy the draft constraint of a segment given by equation (19). The molecule of the chemical reaction optimization algorithm comprises of all the variables which can be depicted in figure 3.

### *Molecules      Variable arrangement in a population*

$$\begin{bmatrix} 1 & q_v^t & \lambda_c^t & h_v^t & a_v^t & f_v^t & d_v^t & \omega_{sv}^t & z_{sv}^t & \theta_{vm}^t & \delta_{sv}^t & y_{vmn}^t & x_{vmt} \\ 2 & q_v^t & \lambda_c^t & h_v^t & a_v^t & f_v^t & d_v^t & \omega_{sv}^t & z_{sv}^t & \theta_{vm}^t & \delta_{sv}^t & y_{vmn}^t & x_{vmt} \\ \cdot & & & & & & & & & & & \cdot & \\ \cdot & & & & & & & & & & & \cdot & \\ \cdot & & & & & & & & & & & \cdot & \\ 100 & q_v^t & \lambda_c^t & h_v^t & a_v^t & f_v^t & d_v^t & \omega_{sv}^t & z_{sv}^t & \theta_{vm}^t & \delta_{sv}^t & y_{vmn}^t & x_{vmt} \end{bmatrix}$$

Figure 3: Arrangement of the variable in molecules of a population

Chemical reaction optimization algorithm comprises of three stages – initialization, iterations and terminations. In initialization stage, the values of different parameters of the algorithm like *PopSize* (number of molecules in a population), *KELossRate* (loss rate of kinetic energy in the elementary reactions) and *MolColl* (determines the type of reaction occurring) are set. Figure 4 presents the pseudo code related to the initialization phase. Values pertaining to *buffer*, *Initial KE*,  $\alpha$  and  $\beta$  are assigned in the initialization phase. The four elementary reactions of the CRO algorithm such as on wall ineffective collision, decomposition, inter molecular ineffective collision and synthesis are described in the following sections.

```

Function Molecule
    Assigning values to  $\omega$ ,  $PE$ ,  $KE$ ,  $NumHit$ ,  $MinStruct$ ,  $MinPE$ ,  $MinHit$ 
Procedure
    Molecule ()
    {
        Randomly create  $\omega$  in the solution space
        Calculate  $PE = f(\omega)$ 
        Set  $KE = InitialKE$ 
        Set  $NumHit = 0$ 
        Set  $MinStruct = \omega$ 
        Set  $MinPE = PE$ 
        Set  $MinHit = 0$ 
    }
    Onwall Ineffective Collision ()
    Decomposition ()
    Intermolecular Ineffective Collision ()
    Synthesis ()
end function

```

Figure 4: Algorithm for molecule initialization

#### 4.2. On-wall ineffective collision

In this reaction, the collision of the molecules takes place with the wall of the container and after the reaction only one single molecule is left. The molecule structure  $\omega$  changes to  $\omega_1$  after the collision and little variation occurs to the potential energy. The change takes place only if,

$$PE_{\omega} + KE_{\omega} \geq PE_{\omega_1} \quad (26)$$

$$KE_{\omega_1} = (PE_{\omega} + KE_{\omega} - PE_{\omega_1}) \times a \quad (27)$$

Here,  $a$  is the random number in the interval of  $[KELossRate, 1]$  and  $(1-a)$  depicts the kinetic energy ( $KE$ ) lost to the surrounding environment due to the collision of the molecule with the wall. The central energy buffer stores the remaining energy which can be used to trigger the decomposition reaction. Although, if the equation (26) is not satisfied, then the on-wall ineffective collision doesn't take place and the original molecule with the same structure is retained within the population. Figure 5 presents the pseudo code pertaining to the on-wall

ineffective collision. Equation (27) helps to compute the kinetic energy of the newly formed molecule using the values of potential energy and kinetic energy of original particle and potential energy of new particles.

```

Function On-wall Ineffective Collision ( $M$ ,  $buffer$ )
  Input: Molecule  $M$  with structure  $\omega$  and central energy buffer
  New molecule is obtained,  $\omega = N(\omega)$ 
   $PE_{\omega}$  is calculated using  $f(\omega)$ , that is  $PE_{\omega} = f(\omega)$ 
  If  $PE_{\omega} + KE_{\omega} \geq PE_{\omega_1}$ 
    Generate  $a$  in the interval  $[KELossRate, 1]$ 
    Set  $KE_{\omega_1} = (PE_{\omega} + KE_{\omega} - PE_{\omega_1}) \times a$ 
    Update  $buffer = buffer + (PE_{\omega} + KE_{\omega} - PE_{\omega_1}) \times (1 - a)$ 
    Molecule  $M$  is updated,  $\omega = \omega_1$ ,  $PE_{\omega} = PE_{\omega_1}$  and  $KE_{\omega} = KE_{\omega_1}$ 
    If  $PE_{\omega} < MinPE_{\omega}$ 
      Update  $MinStruct_{\omega} = \omega$ ,  $MinPE_{\omega} = PE_{\omega}$  and  $MinHit_{\omega} = NumHit_{\omega}$ 
    end
  end
  Output:  $M$  and  $buffer$ 
end function

```

Figure 5: Algorithm for on-wall ineffective collision

### 4.3. Decomposition

In this reaction, a single molecule collides with the wall of the container and decomposes into two molecules and the structures of the newly formed molecules are different from the structure of the original molecule. Decomposition reaction is essential for exploring new promising search spaces after performing enough local searches carried out by on-wall ineffective collision. As new molecules are created in this reaction, hence more energy is required, which can be taken from the central energy buffer, depending on two random numbers  $\delta_1, \delta_2$  uniformly generated in the range (0, 1). For decomposition reaction, the modified energy conservation condition is presented as follows,

$$PE_{\omega} + KE_{\omega} + (\delta_1 \times \delta_2 \times buffer) \geq PE_{\omega_1} + PE_{\omega_2} \quad (28)$$

If the condition given in equation (28) holds true, then the two molecules form using the original molecule and the energy associated with decomposition reaction is given by equation (29).

$$E_{deco} = (PE_{\omega} + KE_{\omega} + (\delta_1 \times \delta_2 \times buffer)) - (PE_{\omega_1''} + PE_{\omega_2''}) \quad (29)$$

$$KE_{\omega_1''} = E_{deco} \times \delta_3 \quad (30)$$

$$KE_{\omega_2''} = E_{deco} \times (1 - \delta_3) \quad (31)$$

$$buffer'' = (1 - \delta_1 \delta_2) buffer \quad (32)$$

Equation (30) and (31) computes the kinetic energy of the newly formed molecules obtained from the energy related to the decomposition reaction. Here,  $\delta_3$  is a randomly generated number in an interval of (0, 1). Equation (32) depicts the energy associated with the central buffer. Figure 6 presents the pseudo code pertaining to the decomposition reaction.

#### 4.4. Inter-molecular ineffective collision

This reaction occurs when two randomly selected molecules collide with each other and as a result two new molecules are formed. The new molecules are formed from the original molecules neighborhood structure without any major change in the molecularity. Inter-molecular ineffective collision is similar to the on-wall ineffective collision from the fact that both the reactions perform as local search operator for CRO. Inter-molecular ineffective collision takes place only when the following condition given in equation (33) is met. The energy released after the reaction can be estimated from the following equation (34).

$$PE_{\omega_1} + PE_{\omega_2} + KE_{\omega_1} + KE_{\omega_2} \geq PE_{\omega_1''} + PE_{\omega_2''} \quad (33)$$

$$E_{inter} = (PE_{\omega_1} + PE_{\omega_2} + KE_{\omega_1} + KE_{\omega_2}) - (PE_{\omega_1''} + PE_{\omega_2''}) \quad (34)$$

$$KE_{\omega_1''} = E_{inter} \times \delta_4 \quad (35)$$

$$KE_{\omega_1} = E_{inter} \times (1 - \delta_4) \quad (36)$$

The kinetic energy pertaining to the two newly formed molecules can be computed using equations (35) and (36) and here random number  $\delta_4$  is generated within an interval of (0, 1). Figure 7 provides the pseudo-code for inter-molecular ineffective collision.

**Function** Decomposition ( $M, buffer$ )**Input:** Molecule  $M$  with structure  $\omega$  and central energy bufferCreating  $M_{\omega_1}''$  and  $M_{\omega_2}''$  $\omega_1''$  and  $\omega_2''$  are obtained from  $\omega$ Calculating  $PE_{\omega_1}'' = f(\omega_1'')$  and  $PE_{\omega_2}'' = f(\omega_2'')$ **If**  $PE_{\omega} + KE_{\omega} \geq PE_{\omega_1}'' + PE_{\omega_2}''$ 

$$E_{deco} = PE_{\omega} + KE_{\omega} - PE_{\omega_1}'' - PE_{\omega_2}''$$

Generate  $\delta_3 \in [0,1]$ Compute  $KE_{\omega_1}'' = E_{deco} \times \delta_3$  and  $KE_{\omega_2}'' = E_{deco} \times (1 - \delta_3)$ Update  $MinStruct_{\omega_1}'' = \omega_1''$  and  $MinStruct_{\omega_2}'' = \omega_2''$ 

$$MinPE_{\omega_1}'' = PE_{\omega_1}'' \text{ and } MinPE_{\omega_2}'' = PE_{\omega_2}''$$

Destroy  $M_{\omega}$ **else**Generate  $\delta_1, \delta_2 \in [0,1]$ 

$$E_{deco} = (PE_{\omega} + KE_{\omega} + (\delta_1 \times \delta_2 \times buffer)) - (PE_{\omega_1}'' + PE_{\omega_2}'')$$

**If**  $E_{deco} \geq 0$ 

$$buffer'' = (1 - \delta_1 \delta_2) buffer$$

Generate  $\delta_3 \in [0,1]$ Compute  $KE_{\omega_1}'' = E_{deco} \times \delta_3$  and  $KE_{\omega_2}'' = E_{deco} \times (1 - \delta_3)$ Update  $MinStruct_{\omega_1}'' = \omega_1''$  and  $MinStruct_{\omega_2}'' = \omega_2''$ 

$$MinPE_{\omega_1}'' = PE_{\omega_1}'' \text{ and } MinPE_{\omega_2}'' = PE_{\omega_2}''$$

Destroy  $M_{\omega}$ **else**Destroy  $M_{\omega_1}''$  and  $M_{\omega_2}''$ **end****end****Output:**  $M$  and  $buffer$ **end function**

Figure 6: Algorithm for decomposition reaction

**Function** Inter-Molecular Ineffective Collision ( $M_{\omega_1}''$  and  $M_{\omega_2}''$ )

**Input:** Molecules  $M_{\omega_1}$  and  $M_{\omega_2}$

New molecules are obtained,  $\omega_1 = N(\omega_1)$  and  $\omega_2 = N(\omega_2)$

Calculating  $PE_{\omega_1} = f(\omega_1)$  and  $PE_{\omega_2} = f(\omega_2)$

Set  $E_{inter} = (PE_{\omega_1} + PE_{\omega_2} + KE_{\omega_1} + KE_{\omega_2}) - (PE_{\omega_1}'' + PE_{\omega_2}'')$

**If**  $E_{inter} \geq 0$

Generate  $\delta_4$  in the interval (0,1)

Calculating  $KE_{\omega_1}'' = E_{inter} \times \delta_4$  and  $KE_{\omega_2}'' = E_{inter} \times (1 - \delta_4)$

Assigning  $\omega_1 = N(\omega_1'')$  and  $\omega_2 = N(\omega_2'')$

Update  $PE_{\omega_1} = PE_{\omega_1}''$  and  $PE_{\omega_2} = PE_{\omega_2}''$

Update  $KE_{\omega_1} = KE_{\omega_1}''$  and  $KE_{\omega_2} = KE_{\omega_2}''$

**If**  $PE_{\omega_1} < MinPE_{\omega_1}$

Update  $MinStruct_{\omega_1} = \omega_1$ ,  $MinPE_{\omega_1} = PE_{\omega_1}$

**end**

**If**  $PE_{\omega_2} < MinPE_{\omega_2}$

Update  $MinStruct_{\omega_2} = \omega_2$ ,  $MinPE_{\omega_2} = PE_{\omega_2}$

**end**

**end**

**Output:**  $M_{\omega_1}''$  and  $M_{\omega_2}''$

**end function**

Figure 7: Algorithm for inter-molecular ineffective collision

#### 4.5. Synthesis

In this reaction, two molecules collide with each other to form a single molecule and synthesis takes place only when the following condition given by equation (37) is met,

$$PE_{\omega_1} + PE_{\omega_2} + KE_{\omega_1} + KE_{\omega_2} \geq PE_{\omega}'' \quad (37)$$

$$KE_{\omega}'' = PE_{\omega_1} + PE_{\omega_2} + KE_{\omega_1} + KE_{\omega_2} - PE_{\omega}'' \quad (38)$$

The energy released after performing the synthesis reaction is given by equation (38). The newly formed molecule has the potential to look more promising solution space because of higher



kinetic energy and thus, this reaction helps in exploration of the search space. Figure 8 presents the pseudo code associated with the synthesis reaction.

```

Function Synthesis ( $M_{\omega''}$ )
  Input: Molecules  $M_{\omega_1}$  and  $M_{\omega_2}$ 
  New molecule is obtained,  $\omega'' = N(\omega_1, \omega_2)$ 
   $PE_{\omega''}$  is calculated using  $f(\omega'')$ , that is  $PE_{\omega''} = f(\omega'')$ 
  If  $PE_{\omega_1} + PE_{\omega_2} + KE_{\omega_1} + KE_{\omega_2} \geq PE_{\omega''}$ 
    Set  $KE_{\omega''} = PE_{\omega_1} + PE_{\omega_2} + KE_{\omega_1} + KE_{\omega_2} - PE_{\omega''}$ 
    Update  $MinStruct_{\omega''} = \omega''$  and  $MinPE_{\omega''} = PE_{\omega''}$ 
    Destroy  $M_{\omega_1}$  and  $M_{\omega_2}$ 
  else
    Destroy  $M_{\omega''}$ 
  end
  Output:  $M_{\omega''}$ 
end function

```

Figure 8: Algorithm for synthesis reaction

#### 4.6. Chemical Reaction Optimization Algorithm Framework

During each iteration of CRO algorithm, a random number  $b$  is generated within a range of (0, 1) and the value of  $b$  determines the occurrence of unimolecular or inter-molecular collision. If  $b > MoleColl$ , then one molecule is selected and if  $b \leq MoleColl$ , then two molecules are considered for carrying out a CRO reaction. Decomposition takes place using a single molecule if  $(NumHit - MinHit) > \alpha$ , otherwise on-wall ineffective collision is performed for  $(NumHit - MinHit) \leq \alpha$ . For two molecules scenario, the synthesis reaction occurs only when the kinetic energy of both the molecules is less than  $\beta$  and inter-molecular collision gets triggered when kinetic energy for both molecules is greater than  $\beta$ . The algorithm evaluates the best solution for iteration and compares it with the global best solution and accordingly updates the global best at the end of the iteration if the current best solution is better than the global best

solution. Once the stopping criteria are met, the algorithm stops and obtains the global best solution as the output of the algorithm. Figure 9 provides the pseudo-code of the chemical reaction optimization and figure 10 presents the flowchart of the CRO algorithm. The stepwise procedure of CRO algorithm is presented as follows,

Step 1: The values associated with the objective function and constraints of the mathematical formulation are taken into consideration.

Step 2: Initialization stage

Step 2.1: Parameter values of the CRO algorithm such as *PopSize*,  $\alpha$ ,  $\beta$ , *buffer*, *MolColl*, *KELossRate* and *KE* are assigned.

Step 2.2: Creating a *PopSize* of number of molecules where each molecule comprises of all the variables of the mathematical formulation presented in section 3.1.

Step 3: Computing the potential energy *PE* of each molecule considering the objective function of the mathematical model.

Step 4: If the stopping criteria (maximum iteration) is satisfied, then the algorithm is terminated, otherwise, step 4.1 to 4.3 are executed.

Step 4.1: Randomly generating *b* within the range (0, 1) and if  $b > \text{MoleColl}$ , then step 4.2 is executed. Although, step 4.3 is executed if  $b \leq \text{MoleColl}$ .

Step 4.2: A single molecule is randomly selected from the *PopSize* for performing the decomposition if  $(\text{NumHit} - \text{MinHit}) > \alpha$ . Although, if  $(\text{NumHit} - \text{MinHit}) \leq \alpha$ , then on-wall ineffective collision is carried out using the selected molecule.

Step 4.3: Two molecules are randomly selected from the *PopSize* for implementing the inter-molecular ineffective collision based on the condition  $KE < \beta$ . Although, if  $KE \geq \beta$ , then synthesis is carried out on the two selected molecules.

Step 5: The best solution of the iteration is obtained and compared with the global best solution.

Step 6: Iteration continues and goes back to step 4.

**Function** Chemical reaction optimization algorithm

**Input:** Information pertaining to the problem like objective function, constraints and parameter values are provided to the algorithm

Assigning values to algorithmic parameters such as *PopSize*,  $\alpha$ ,  $\beta$ , *buffer*, *InitialKE*, *MolColl* and *KELossRate*

Using the algorithm for molecule initialization given in figure 3, a population containing number of molecules is created

**While** stopping criteria (total number of iteration is not reached) is not satisfied

Randomly generate  $b$  within  $[0, 1]$

**If**  $b > MoleColl$

A single molecule  $M_{\omega}$  is randomly selected from the *PopSize*

**If** Decomposition criteria is satisfied

Decomposition is performed using its algorithm given in figure 5

**else**

Executing on-wall ineffective collision using the algorithm in figure 4

**end**

**else**

Two molecules  $M_{\omega_1}$  and  $M_{\omega_2}$  are randomly selected from *PopSize*

**If** Synthesis criteria is satisfied

Synthesis reaction is performed using the algorithm given in figure 7

**else**

Executing Inter-molecular ineffective collision using the algorithm given in figure 6

**end**

**end**

Best solution of the iteration is obtained and compared with the global best

**Output:** Global best solution and its objective function value

**end function**

Figure 9: Algorithm for chemical reaction optimization function

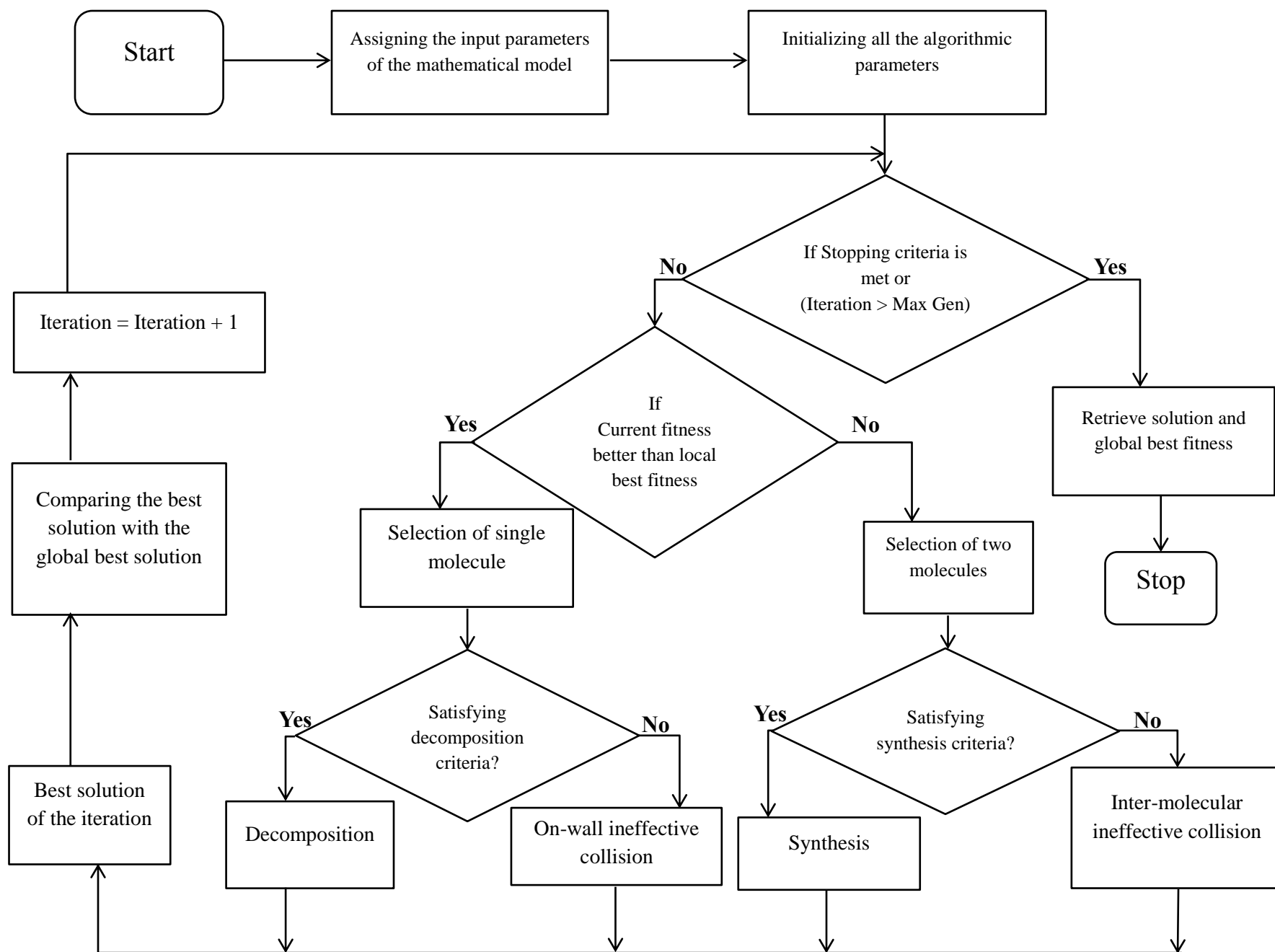


Figure 10: Flowchart of chemical reaction optimization algorithm

## 5. Results and Discussion

In this section, the proposed discrete hybrid berth allocation problem is solved using chemical reaction optimization algorithm and the obtained solutions are compared with genetic algorithm. The algorithm is coded on window 7 system with Intel core-i-5 2.4 GHZ and 4 GB of RAM in MATLAB R2015a Software.

### 5.1. Parameter settings and Computational Experiment

The chemical reaction optimization algorithm is employed to resolve the problem presented in section 3.1. The computational experiment is conducted to find the near-optimum solution using CRO, BBGA, GA and PSO algorithms. It is necessary to obtain the ideal parameter settings in order to ensure superior performance of all the algorithms. A set of preliminary experiments are conducted for finding the appropriate parameter setting that produces overall good results across most instances. Based on the preliminary experiments undertaken, the best values of the parameters for CRO are: *MolColl* = 0.12, *buffer* = 0, *KELossRate* = 0.21,  $\alpha = 50$ ,  $\beta = 12$ , *Generations* = 200 and for GA; the crossover probability ( $p_c = 0.48$ ) and mutation probability ( $p_m = 0.16$ ) and population = 200. For PSO, the values of the parameter are given as follows, inertia weight = 0.9, acceleration coefficients = 0.1, 0.98 and swarm size = 200. For more information regarding the parameter values of PSO, please refer to De et al. (2017a). For BBGA, the parameter values are, mutation rate = 1, population = 200, chromosome number = 20 and elite preservation = 0.01. For more information regarding the BBGA algorithm, refer to Chang et al. (2013). The parameters of the algorithms are appropriately tuned to obtain near optimal solutions for each of the problem instances designed. Repeated test trails have been carried out to predict the most appropriate values corresponding to all the parameters for every algorithm. Each instance is run for 30 times and the best solution is provided. For the validation, the model is tested on 7 problem instances as described in table 1 and the performance of the chemical reaction optimization algorithm is validated on all the instances and the results obtained are compared with BBGA, GA and PSO. Table 1 presents the different problem instances based on

the number of vessels, number of segments, container group and number of time periods. The complexities of the problem instances can be realized after observing the total number of variables and constraints given in table 1. Table 2 provides the results obtained after employing CRO, BBGA, GA and PSO algorithm on the problem instances. By varying the berth segments, the mathematical model is tested for ports of varying quay length and ports with the capability of allowing several vessels to perform their loading/unloading operation. The results provide the total cost (TC) including total service costs, fuel cost and penalty cost due to the delay in starting port operations. Table 2 also presents the exact result for the first problem instance (15, 10, 2, 10) obtained using LINGO. Due to memory restrictions, the exact solution pertaining to other problem instances could not be obtained.

Table 1: Different problem instances considered for the computational experiment

Problem instance (vessel, segment container group, time period)	Number of binary variables	Number of continuous variables	Total number of variables	Number of inequality constraints	Number of equality constraints	Total number of constraints
(15, 10, 2, 10)	24,750	770	25,520	11,415	3,450	14,865
(30, 20, 2, 30)	477,000	4560	481,560	131,430	38,700	170,130
(49, 40, 5, 30)	2, 685,690	7500	2, 693,190	355,789	122,010	477,799
(59, 65, 10, 35)	9, 358,580	10,675	9, 369,255	634,014	274,645	908,659
(65, 65, 10, 40)	11, 830,000	13,400	11, 843,400	860,665	345,800	1, 206,465
(78, 75, 15, 50)	23, 435,100	20,250	23, 455,350	1, 532,778	596,700	2, 129,478
(82, 85, 15, 50)	31, 328,100	21,250	31, 349,350	1, 717,982	709,300	2, 427,282

Table 2: Results obtained for different problem instances using CRO, BBGA, GA and PSO

Problem instance (vessel, segment container group, time period)	CRO results		BBGA results		GA results		PSO results		Exact results
	Total Cost (Rs.)	Operational Time (hrs.)	Total Cost (Rs.)	Operational Time (hrs.)	Total Cost (Rs.)	Operational Time (hrs.)	Total Cost (Rs.)	Operational Time (hrs.)	Total Cost (Rs.)
(15, 10, 2, 10)	2, 79,454	875.64	2, 84,161	901.33	2, 88,585	938.72	2, 87,616	932.14	2, 72,197
(30, 20, 2, 30)	8, 71,485	1314.11	8, 93,578	1378.42	9, 04,746	1391.66	9, 02,166	1388.57	-
(50, 40, 5, 30)	12, 66,368	2160.27	12, 72,489	2172.10	12, 75,398	2178.18	12, 74,491	2175.60	-
(59, 65, 10, 35)	13, 82,452	3051.86	13, 96,278	3196.14	13, 98,362	3198.29	13, 96,112	3195.82	-
(65, 65, 10, 40)	14, 66,293	3985.73	14, 79,661	4076.19	14, 81,227	4112.58	14, 81,107	4098.26	-
(78, 75, 15, 50)	16, 01,214	4371.39	16, 51,107	4429.72	16, 57,093	4432.65	16, 56,113	4429.91	-
(82, 85, 15, 50)	17, 26,523	4819.32	17, 49,886	4875.11	17, 55,663	4903.25	17, 53,907	4886.89	-

## 5.2. Efficacy of the algorithms

Table 3 presents the mean and standard-deviation values as well as the best and worst solutions obtained after carrying out 30 runs on each algorithm for solving the problem instances. Mean and the standard deviation value of CRO for all the problem sizes are much better than other algorithms such as BBGA, GA and PSO which automatically proves the superiority of the CRO algorithm. CRO outperforms the other algorithms as it has the better exploration capability in terms of effectively carrying out the searching procedure using its different elementary reactions such to attain promising near optimal solutions. CRO uses different elementary reactions such as on wall ineffective collision, decomposition, inter molecular ineffective collision and synthesis. Decomposition and synthesis operator of CRO is responsible for the exploration of new search space and it overcomes overcome many entrapments when the local diversity of the particles decreases. On-wall ineffective collision helps to carry out the local search procedure for CRO to explore promising solution spaces where as other benchmark algorithms like PSO and GA suffer from premature convergence or getting confined to a local solution. BBGA still provides better solution than PSO and GA, although, CRO easily outperforms BBGA at it employs advanced searching capabilities making it more powerful for overcoming many challenging scenarios and attaining better near-optimal solutions.

Table 3: Computational results justifying the efficacy of the algorithms

Problem instance (vessel, segment container group, time period)	Algorithm	Best solution	Worst solution	Mean	Standard deviation
Problem Instance 1 (15, 10, 2, 10)	CRO	$2.794 \times 10^5$	$2.891 \times 10^5$	$2.832 \times 10^5$	$1.546 \times 10^4$
	BBGA	$2.841 \times 10^5$	$2.974 \times 10^5$	$2.926 \times 10^5$	$2.557 \times 10^4$
	GA	$2.885 \times 10^5$	$2.987 \times 10^5$	$2.941 \times 10^5$	$2.811 \times 10^4$
	PSO	$2.876 \times 10^5$	$2.976 \times 10^5$	$2.935 \times 10^5$	$2.849 \times 10^4$
Problem Instance 2 (30, 20, 2, 30)	CRO	$8.714 \times 10^5$	$8.903 \times 10^5$	$8.809 \times 10^5$	$3.154 \times 10^4$
	BBGA	$8.935 \times 10^5$	$9.181 \times 10^5$	$9.136 \times 10^5$	$4.551 \times 10^4$
	GA	$9.047 \times 10^5$	$9.275 \times 10^5$	$9.224 \times 10^5$	$4.824 \times 10^4$
	PSO	$9.021 \times 10^5$	$9.263 \times 10^5$	$9.217 \times 10^5$	$4.743 \times 10^4$
Problem Instance 3 (49, 40, 5, 30)	CRO	$1.266 \times 10^6$	$1.285 \times 10^6$	$1.270 \times 10^6$	$5.418 \times 10^4$
	BBGA	$1.272 \times 10^6$	$1.301 \times 10^6$	$1.293 \times 10^6$	$6.745 \times 10^4$
	GA	$1.275 \times 10^6$	$1.319 \times 10^6$	$1.306 \times 10^6$	$7.138 \times 10^4$
	PSO	$1.274 \times 10^6$	$1.315 \times 10^6$	$1.304 \times 10^6$	$7.011 \times 10^4$
Problem Instance 4	CRO	$1.382 \times 10^6$	$1.424 \times 10^6$	$1.398 \times 10^6$	$7.187 \times 10^4$
	BBGA	$1.396 \times 10^6$	$1.468 \times 10^6$	$1.438 \times 10^6$	$8.278 \times 10^4$



(59, 65, 10, 35)	GA	$1.398 \times 10^6$	$1.485 \times 10^6$	$1.452 \times 10^6$	$8.541 \times 10^4$
	PSO	$1.396 \times 10^6$	$1.479 \times 10^6$	$1.446 \times 10^6$	$8.475 \times 10^4$
Problem Instance 5 (65, 65, 10, 40)	CRO	$1.466 \times 10^6$	$1.522 \times 10^6$	$1.489 \times 10^6$	$8.012 \times 10^4$
	BBGA	$1.479 \times 10^6$	$1.568 \times 10^6$	$1.535 \times 10^6$	$8.944 \times 10^4$
	GA	$1.481 \times 10^6$	$1.579 \times 10^6$	$1.542 \times 10^6$	$9.123 \times 10^4$
	PSO	$1.481 \times 10^6$	$1.574 \times 10^6$	$1.538 \times 10^6$	$9.074 \times 10^4$
Problem Instance 6 (78, 75, 15, 50)	CRO	$1.601 \times 10^6$	$1.668 \times 10^6$	$1.631 \times 10^6$	$8.817 \times 10^4$
	BBGA	$1.651 \times 10^6$	$1.731 \times 10^6$	$1.689 \times 10^6$	$9.918 \times 10^4$
	GA	$1.657 \times 10^6$	$1.754 \times 10^6$	$1.712 \times 10^6$	$1.074 \times 10^5$
	PSO	$1.656 \times 10^6$	$1.751 \times 10^6$	$1.709 \times 10^6$	$1.058 \times 10^5$
Problem Instance 7 (82, 85, 15, 50)	CRO	$1.726 \times 10^6$	$1.783 \times 10^6$	$1.769 \times 10^6$	$1.114 \times 10^5$
	BBGA	$1.749 \times 10^6$	$1.864 \times 10^6$	$1.815 \times 10^6$	$1.244 \times 10^5$
	GA	$1.755 \times 10^6$	$1.897 \times 10^6$	$1.846 \times 10^6$	$1.306 \times 10^5$
	PSO	$1.753 \times 10^6$	$1.889 \times 10^6$	$1.838 \times 10^6$	$1.295 \times 10^5$

### 5.3. Case study

The model is test on a real-world case study of a port based on the eastern region of India. The following assumptions are considered to for the case study. A ship can occupy a maximum of sixteen segments and minimum of fourteen segments of the berth. The berth segment is divided into 64 sections; each section is of 17.7 meters' length. The container-ship has only two types of containers 20 ft. and 40 ft. shipping containers. The characteristics of the studied port are as follows: container capacity of 1.25 million TEU/annum and berth line for international container terminal has a length of 1140 m with 12.5 m to 12.9 m draft length. The model is tested for a scenario where 49 ships arrived at the anchorage of the port during the month of September 2016. The port manager/ authorities has the information about some of the following parameters, i.e. the number of containers on the ships, ship size, and expected arrival time of ships at the anchorage, available space and position for containers.

### 5.4. Input Data and Result and Analysis

The ships arrive at the port and search the best possible location for unloading the container to the stockyard. In order to determine the optimal berth allocation for arrived ship, Chemical reaction optimization, Block based Genetic Algorithm, Genetic algorithm and Particle Swarm Optimization are used to test the practical scenarios of a real-world port located in the eastern part of India. The table 4 describes the ship expected arrival time at the port, number of containers available on the ship, container type (20 ft. or 40 ft.). The maximum numbers of quay

cranes available at the berth are 5 and the maximum unloading rate of each quay crane is 72 containers per hour. Port can handle two types of Ships (Panamax and Capemax)) and different instances of the scenario is described in Table 4.

For the sake of illustration, the result of a case study is presented in table 5. The berthing day and time, departure day and time and total operating time of each vessel is provided in table 5. In order to show the comparative result, the convergence graph of CRO, BBGA, GA and PSO is provided showcasing the total operational time with respect to number of generations. Figures 11(a), 11(b), 11(c) and 11(d) provide the convergence graphs for the algorithms CRO, BBGA, GA and PSO respectively. From the graph, it is amply clear that the GA and PSO converges early, whereas CRO and BBGA requires more iteration for converging as it extensively carries out the searching procedure.

Table 4: Ships Arrival information at the Anchorage of the port

Ship no	Exp. Arr. date	Exp. Arr. month	Exp. Arr. time	No. of containers	Cont. cap.	Ship size
1	3	9	15.10	10500	20 ft.	P
2	3	9	19.50	4500	20 ft.	C
3	4	9	14.20	5101	20 ft.	C
4	5	9	11.20	4800	20 ft.	C
5	5	9	12.15	10100	20 ft.	C
6	5	9	16.10	3500	20 ft.	C
7	6	9	16.50	12360	20 ft.	P
8	7	9	10.75	10800	20 ft.	P
9	7	9	12.06	4950	20 ft.	C
10	7	9	16.20	5051	20 ft.	C
11	8	9	11.45	14020	20 ft.	P
12	8	9	15.70	14050	20 ft.	P
13	10	9	14.15	5020	20 ft.	P
14	11	9	15.20	8560	20 ft.	P
15	11	9	18.10	3001	20 ft.	C
16	11	9	19.30	3856	20 ft.	C
17	12	9	21.75	5210	20 ft.	P
18	12	9	16.50	9520	20 ft.	P
19	12	9	18.75	14200	20 ft.	P
20	13	9	14.50	13110	20 ft.	P
21	14	9	15.25	5101	20 ft.	C
22	14	9	10.50	3001	20 ft.	C
23	15	9	11.58	4520	20 ft.	P
24	16	9	10.06	14020	20 ft.	P
25	16	9	14.55	10020	20 ft.	P
26	16	9	17.30	5001	20 ft.	C
27	16	9	21.50	14050	20 ft.	P

28	17	9	14.08	3001	40 ft.	C
29	17	9	17.52	12305	20 ft.	P
30	18	9	11.75	8560	20 ft.	P
31	19	9	17.45	14050	20 ft.	P
32	21	9	9.10	5101	20 ft.	C
33	21	9	10.50	3850	20 ft.	C
34	21	9	15.30	3101	20 ft.	C
35	23	9	14.10	3220	20 ft.	C
36	23	9	17.50	14105	20 ft.	P
37	24	9	11.50	4020	20 ft.	C
38	25	9	12.30	5001	20 ft.	C
39	25	9	16.75	14025	20 ft.	P
40	26	9	15.10	3260	20 ft.	C
41	26	9	18.75	14010	20 ft.	P
42	26	9	19.50	5201	20 ft.	C
43	27	9	10.35	13200	20 ft.	P
44	27	9	11.15	4965	20 ft.	C
45	28	9	17.50	3250	20 ft.	C
46	28	9	19.10	10001	20 ft.	P
47	29	9	11.50	2508	20 ft.	C
48	30	9	11.45	4520	20 ft.	C
49	30	9	12.10	4005	20 ft.	C

Table 5: Computational Analysis (CRO)

Ship no	Actual berthing date	Actual berthing month	Actual berthing time	No. of containers	Departure date	Departure time	Operational time
1	3	9	10.15	10500	5	11.93	48.61
2	4	9	10.00	4500	5	22.10	31.27
5	5	9	10.00	10100	7	9.75	47.75
3	5	9	11.30	5101	6	23.50	52.09
4	6	9	10.00	4800	7	19.33	33.3
6	6	9	11.15	3500	7	11.35	24.31
9	6	9	16.42	4950	8	2.42	34.38
7	7	9	10.45	12360	10	0.45	62.54
10	7	9	12.15	5051	8	11.30	23.35
8	8	9	10.30	10800	10	12.30	50.1
11	8	9	11.15	3001	10	7.45	20.84
13	10	9	10.00	5020	11	22.45	34.86
14	10	9	13.86	8560	12	21.55	39.63
12	11	9	10.01	14050	13	5.10	65.05

15	12	9	12.82	5001	14	10.46	34.73
18	12	9	17.45	9520	15	11.51	66.11
16	13	9	10.45	3856	14	13.55	27.16
19	13	9	15.30	14200	17	17.25	98.61
17	14	9	10.15	5210	15	22.35	36.18
20	14	9	12.45	13110	16	23.55	61.2
22	15	9	10.35	3001	16	6.85	20.84
21	15	9	12.16	5101	16	22.1	33.43
23	16	9	10.20	4520	17	16.74	31.38
25	17	9	10.05	10020	19	8.42	46.39
24	17	9	17.80	14020	20	7.18	64.91
28	18	9	10.45	3001	19	21.87	34.6
26	19	9	10.05	5001	20	9.34	23.15
27	19	9	13.15	14050	22	13.82	74.415
30	20	9	10.25	8560	22	22.05	59.44
29	21	9	11.48	12305	24	23.79	85.45
33	22	9	15.05	3850	23	17.15	26.12
31	23	9	10.05	14050	26	10.67	74.29
34	23	9	18.20	3101	24	15.58	21.53
32	24	9	10.35	5101	25	9.18	23.61
35	24	9	17.20	3220	25	15.56	22.36
37	24	9	13.12	4020	24	17.03	27.91
36	25	9	10.10	14105	29	10.45	97.94
40	25	9	17.65	3260	26	16.22	22.63
39	26	9	10.05	14025	29	0.58	64.93
38	26	9	12.82	5001	28	1.65	34.72
42	27	9	10.40	5201	29	11.10	53.7
44	27	9	14.48	4965	29	2.95	34.47
41	28	9	10.05	14010	1	2.91	64.86
43	29	9	10.30	13200	2	14.12	75.92
45	30	9	15.45	3250	1	6.22	36.77
46	1	10	10.05	10001	3	8.35	46.3
49	2	10	15.45	4005	3	19.26	27.81
47	3	10	10.05	2508	4	5.46	17.41
48	3	10	11.84	4520	4	8.76	20.92

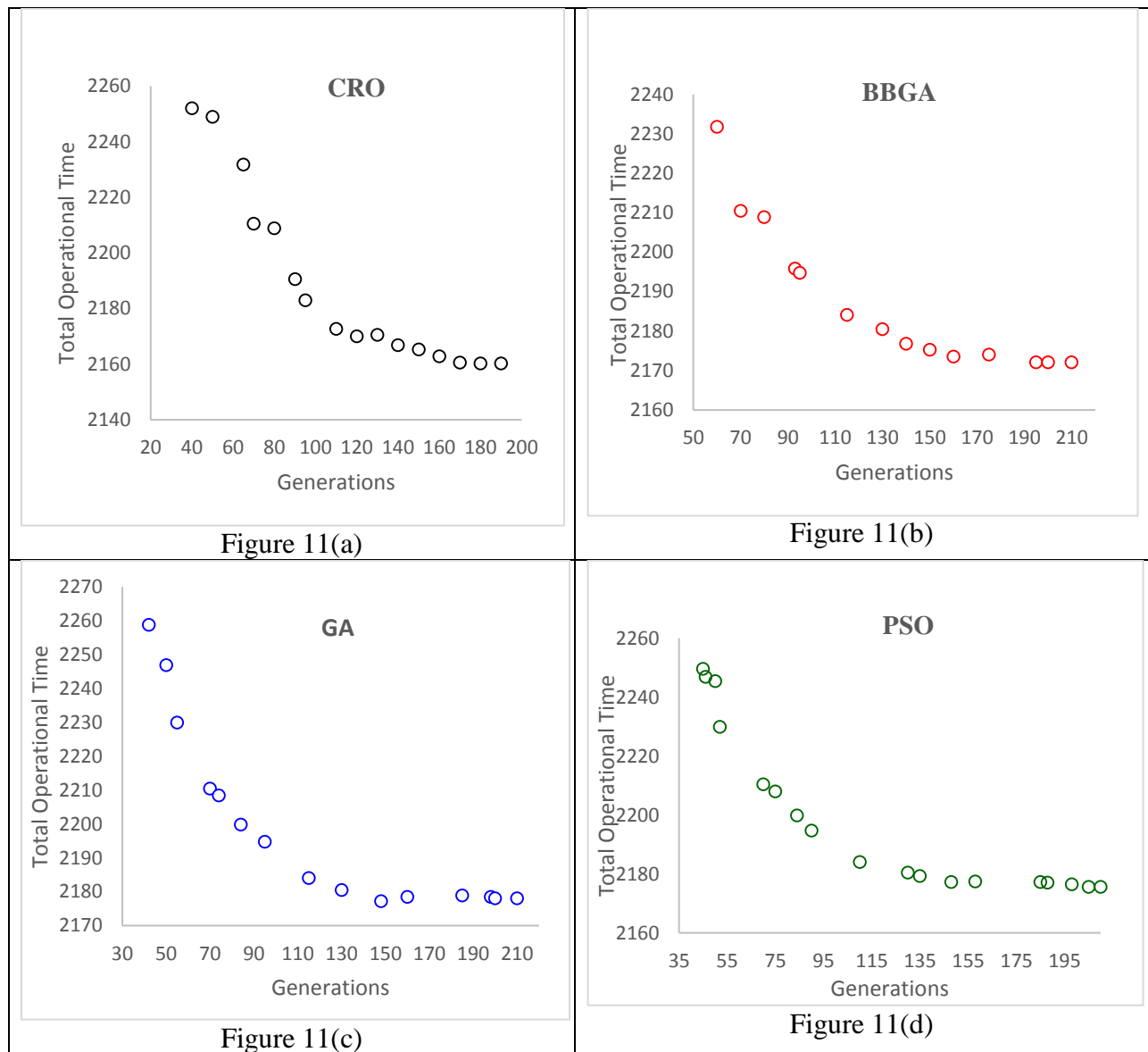


Figure 11: Performance graph of CRO, BBGA, GA and PSO with respect to generation

## 6. Conclusion

The proposed research encounters the berth allocation with a quay crane assignment problem taking into account the vessel arrival and departure information, quay crane information, container handling time, and waiting time of ships. The model is implemented and validated on a

real case scenario of a port situated in the eastern coast of India.

The considered complex scenario of port is formulated as a mixed integer non-linear programming model (MINLP) to minimize the total cost associated with the operating time and fuel consumption. In order to solve the model, a Chemical reaction optimization (CRO) algorithm is introduced and compared with Block-Based Genetic Algorithm (BBGA), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The computational experiment reveals that the proposed Chemical reaction optimization (CRO) algorithm performs better than BBGA, GA and PSO in terms of the total cost and the total operational time during container handling.

This study can be also further extended and synchronized with yard truck allocation to fulfill the customer demand within promising time. The efficacy of the overall port system could be more improved and generate more revenue.

## References

- Babu, S A K I., Pratap. S., Lahoti, G., Fernandes, K J., Tiwari, M K., Mount, M and Xiong, Y. (2015). Minimizing delay of ships in bulk terminals by simultaneous ship scheduling, stockyard planning and train scheduling. *Maritime Economics & Logistics*. 17(4), 464-492.
- Bhattacharjee, K., Bhattacharya, A., and Dey, S. H. N. (2014). Real coded chemical reaction based optimization for short-term hydrothermal scheduling. *Applied Soft Computing*, 24, 962-976.
- Chang, P. C., Chen, M. H., Tiwari, M. K., & Iquebal, A. S. (2013). A block-based evolutionary algorithm for flow-shop scheduling problem. *Applied Soft Computing*, 13(12), 4536-4547.
- Cheong, C. Y., & Tan, K. C. (2008). A multi-objective multi-colony ant algorithm for solving the berth allocation problem. In Y. Liu, A. Sun, H. T. Loh, W. F. Lu, & E. P. Lim (Eds.), *Advances of Computational Intelligence in Industrial Systems* (pp. 333–350). Berlin: Springer-Verlag.
- Cheong, C. Y., Tan, K. C., Liu, D. K., & Lin, C. J. (2010). Multi-objective and prioritized berth allocation in container ports. *Annals of Operations Research*, 180(1), 63-103.
- Christiansen, M., Fagerholt, K., Nygreena, B., Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*. 228 (3), 467–483.

- De, A., Awasthi, A., & Tiwari, M. K. (2015). Robust formulation for optimizing sustainable ship routing and scheduling problem. *IFAC-PapersOnLine*, 48(3), 368-373.
- De, A., Mamanduru, V K R., Gunasekaran, A., Subramanian, N., Tiwari, MK. (2016). Composite particle algorithm for sustainable integrated dynamic ship routing and scheduling optimization. *Computers & Industrial Engineering*. 96, 201-215.
- De, A., Kumar, S. K., Gunasekaran, A., & Tiwari, M. K. (2017a). Sustainable maritime inventory routing problem with time window constraints. *Engineering Applications of Artificial Intelligence*, 61, 77-95.
- De, A., Choudhary, A., & Tiwari, M. K. (2017b). Multiobjective Approach for Sustainable Ship Routing and Scheduling With Draft Restrictions. *IEEE Transactions on Engineering Management*.
- Golias, M. M., Saharidis, G. K., Boile, M., Theofanis, S., & Ierapetritou, M. G. (2009). The berth allocation problem: Optimizing vessel arrival time. *Maritime Economics & Logistics*, 11(4), 358-377.
- Guan, Y., and Cheung, R. K. (2004). The berth allocation problem: Models and solution methods. *OR Spectrum*, 26, 75–92.
- Guan, J., and Lin, G. (2016). Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem. *European Journal of Operational Research*, 248(3), 899–909.
- Hendriks, M. P.M., Lefebvre, E., & Udding, J. T. (2013). Simultaneous berth allocation and yard planning at tactical level. *OR Spectrum*, 35, 441–456.
- Hansen, P., Oğuz, C., Mladenović, N. (2008). Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*. 191 (3), 636–649.
- Hu, Q. M., Hu, Z. H., and Du, Y. (2014). Berth and quay-crane allocation problem considering fuel consumption and emissions from vessels. *Computers & Industrial Engineering*, 70, 1-10.
- Imai, A., Nagaiwa, K. and Tat, C. W. (1997). Efficient planning of berth allocation for container terminals in Asia. *Journal of Advanced Transportation*. 31 (1), 75-94.
- Imai, A., Nishimura, E. and Papadimitriou, S. (2001). The dynamic berth allocation for a container port. *Transportation Research Part B*, 35, pp. 401–417.
- Imai, A., Nishimura, E., Hattori, M., & Papadimitriou, S. (2007). Berth allocation at indented berths for mega-containerships. *European Journal of Operational Research*, 179, 579–593.

- Iris, C., Pacino, D., Ropke, S., Larsen, A. (2015). Integrated Berth Allocation and Quay Crane Assignment Problem: Set partitioning models and computational results. *Transportation Research Part E: Logistics and Transportation Review*. 81, 75–97.
- Kim, K.H and Moon, K.C. (2003). Berth scheduling by simulated annealing. *Transportation Research Part B*, 37, 541–560.
- Lalla-Ruiz, E., González-Velarde, J. L., Melián-Batista, B., & Moreno-Vega, J. M. (2014). Biased random key genetic algorithm for the tactical berth allocation problem. *Applied Soft Computing*, 22, 60-76.
- Lalla-Ruiz, E., Expósito-Izquierdo, C., Melián-Batista, B & Moreno-Vega, J.M. (2016). A Set-Partitioning-based model for the Berth Allocation Problem under Time-Dependent Limitations. *European Journal of Operational Research*. 250, 1001–1012.
- Lam, A. Y., Li, V. O., & James, J. Q. (2012). Real-coded chemical reaction optimization. *IEEE Transactions on Evolutionary Computation*, 16(3), 339-353.
- Lang, N., & Veenstra, A. (2010). A quantitative analysis of container vessel arrival planning strategies. *OR spectrum*, 32(3), 477-499.
- Lee, D. H., Chen, J. H., & Cao, J. X. (2010). The continuous berth allocation problem: A greedy randomized adaptive search solution. *Transportation Research Part E: Logistics and Transportation Review*, 46(6), 1017-1029.
- Li, J. Q., Pan, Q. K. (2012). Chemical-reaction optimization for flexible job-shop scheduling problems with maintenance activity. *Applied soft computing*, 12(9), 2896-2912.
- Li, J. Q., Pan, Q. K., and Wang, F. T. (2014). A hybrid variable neighborhood search for solving the hybrid flow shop scheduling problem. *Applied Soft Computing*, 24, 63-77.
- Li, Z., Nguyen, T. T., Chen, S., and Truong, T. K. (2015). A hybrid algorithm based on particle swarm and chemical reaction optimization for multi-object problems. *Applied Soft Computing*, 35, 525-540.
- Lim, A. (1998). The berth planning problem. *Operations Research Letters*, 22, 105–110.
- Moorthy, R., & Teo, C. P. (2006). Berth management in container terminal: The template design problem. *OR Spectrum*, 28, 495–518.
- Mauri, G. R., Ribeiro, G. M., Lorena, L. A. N., & Laporte, G. (2016). An adaptive large neighborhood search for the discrete and continuous Berth allocation problem. *Computers & Operations Research*, 70, 140-154.



- Nishimura, E., Imai, A., Papadimitriou, S., 2001. Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research* 131 (2), 282–292.
- Oliveira, R.M. De., Mauri, G.R., Lorena, L.A.N. (2012). Clustering Search for the Berth Allocation Problem. *Expert Systems with Applications*. 39 (5), 5499–5505.
- Pratap, S., B, Manoj Kumar, Cheikhrouhou, N., Tiwari, M.K. (2015a). "The robust quay crane allocation for a discrete bulk material handling port," 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 2015, pp. 1174-1178. doi: 10.1109/IEEM.2015.7385833
- Pratap, S., Nayak, A., Cheikhrouhou, N., Tiwari, M.K. (2015b). Decision Support System for Discrete Robust Berth Allocation. *IFAC-PapersOnLine*, 48 (3), 875-880.
- Ribeiro, G.M., Mauri, G.R., Beluco, S de C., Lorena, L.A.N., Laport, G. (2016.) Berth allocation in an ore terminal with demurrage, despatch and maintenance. *Computers & Industrial Engineering*, 96, 8-15.
- Roy, P. K., Bhui, S., and Paul, C. (2014). Solution of economic load dispatch using hybrid chemical reaction optimization approach. *Applied Soft Computing*, 24, 109-125.
- Tao, Y., and Lee, C-Y. (2015). Joint planning of berth and yard allocation in transshipment terminals using multi-cluster stacking strategy. *Transportation Research Part E: Logistics and Transportation Review*. 83, 34–50.
- Ting, C. J., Wu, K. C., & Chou, H. (2014). Particle swarm optimization algorithm for the berth allocation problem. *Expert Systems with Applications*, 41(4), 1543-1550.
- Tong, C. J., Lau, H. C., & Lim, A. (1999). Ant colony optimization for the ship berthing problem. *Lecture Notes in Computer Science*, 1742, 359–370.
- Truong, T. K., Li, K., and Xu, Y. (2013). Chemical reaction optimization with greedy strategy for the 0–1 knapsack problem. *Applied Soft Computing*, 13(4), 1774-1780.
- Umang, N., Bierlaire, M., & Vacca, I. (2013). Exact and heuristic methods to solve the berth allocation problem in bulk ports. *Transportation Research Part E: Logistics and Transportation Review*, 54, 14-31.
- UNCTAD and WTO, Review of Maritime Transport, 2015, (2015).
- Zhen, Lu. (2015). Tactical berth allocation under uncertainty. *European Journal of Operational Research*. 247 (3), 928–944.

Zhen, L., Yu, S., Wang, S and Zhuo, S. (2016). Scheduling quay cranes and yard trucks for unloading operations in container ports. *Annals of Operation research*. doi:10.1007/s10479-016-2335-9.

Wu, L and Ma, W. (2017). Quay crane scheduling with draft and trim constraints. *Transportation Research Part E: Logistics and Transportation Review*. doi.org/10.1016/j.tre.2016.10.011